What is claimed is:

1. A method of managing a plurality of software development environments coupled to one another through a cross development environment, the method comprising:

detecting an update made in a first software development environment among the plurality of software development environments; and

dynamically modifying a mapping between the first software development environment and a second software development environment among the plurality of software development environments in response to the detected update.

2. The method of claim 1, wherein each of the first and second software development environments comprises a problem tracking tool, and wherein the mapping is configured to convert a problem report generated by the problem tracking tool in the first development environment to an acceptable format for the problem tracking tool in the second development environment.

3. The method of claim 1, wherein the mapping is configured to convert a source code file generated by the first development environment to an acceptable format for the second development environment.

4. The method of claim 1, wherein detecting the update comprises receiving a notification from the first software development environment.

5. The method of claim 1, further comprising notifying an administrator of the cross development environment in response to detecting the update.

6. The method of claim 1, further comprising notifying an administrator of the second software development environment in response to detecting the update.

7. The method of claim 6, further comprising updating the second software development environment in response to notification of the administrator.

8. The method of claim 1, wherein the update comprises an update to content stored in the first software development environment.

9. The method of claim 8, wherein the update comprises an update to at least one of a product, component and release stored in a library repository in the first software development environment.

10. The method of claim 1, wherein the update comprises an update to at least one of a tool, a parameter and a value in the first software development environment.

11. The method of claim 1, wherein the mapping is defined in a mapping data structure comprising a plurality of mapping entries, wherein at least one mapping entry includes a wildcard.

12. The method of claim 1, further comprising transforming a transaction generated by the first software development environment into a format compatible with the second software development environment using the mapping.

13. The method of claim 12, wherein transforming the transaction includes routing the transaction to one of a plurality of cross development environment processes.

14. The method of claim 13, wherein routing the transaction is performed by a router process, the router process configured to perform at least one of failover and load balancing in connection with routing the transaction to a cross development environment process.

15. The method of claim 12, wherein transforming the transaction includes communicating the transaction to the second software development environment, the method further comprising retrying communication of the transaction to the second software development environment in response to unavailability of the second software development environment.

16. A method of managing a plurality of software development environments coupled to one another through a cross development environment, the method comprising:

updating a first software development environment among the plurality of software development environments; and

notifying the cross development environment of the update made in the first software development environment in response to the update.

17. The method of claim 16, further comprising dynamically modifying a mapping between the first software development environment and a second software development environment among the plurality of software development environments in response to notification of the cross development environment.

18. The method of claim 16, further comprising dynamically notifying an administrator in response to the update.

19. An apparatus, comprising:

a memory configured to store a mapping data structure for use in a cross development environment that couples together a plurality of software development environments;

a processor; and

program code configured to detect an update made in a first software development environment among the plurality of software development environments, and dynamically modify the mapping data structure to modify a mapping between the first software development environment and a second software development environment among the plurality of software development environments in response to the detected update.

20. The apparatus of claim 19, wherein each of the first and second software development environments comprises a problem tracking tool, and wherein the mapping data structure is configured to convert a problem report generated by the problem tracking tool in the first development environment to an acceptable format for the problem tracking tool in the second development environment.

21. The apparatus of claim 19, wherein the mapping data structure is configured to convert a source code file generated by the first development environment to an acceptable format for the second development environment.

22. The apparatus of claim 19, wherein the program code is configured to detect the update by receiving a notification from the first software development environment.

23. The apparatus of claim 19, wherein the program code is further configured to notify an administrator of the cross development environment in response to detecting the update.

24. The apparatus of claim 19, wherein the program code is further configured to notify an administrator of the second software development environment in response to detecting the update.

25. The apparatus of claim 19, wherein the update comprises an update to content stored in the first software development environment.

26. The apparatus of claim 25, wherein the update comprises an update to at least one of a product, component and release stored in a library repository in the first software development environment.

27. The apparatus of claim 19, wherein the update comprises an update to at least one of a tool, a parameter and a value in the first software development environment.

28. The apparatus of claim 19, wherein the mapping data structure includes a plurality of mapping entries, wherein at least one mapping entry includes a wildcard.

29. The apparatus of claim 19, wherein the program code is further configured to transform a transaction generated by the first software development environment into a format compatible with the second software development environment using the mapping data structure.

30. The apparatus of claim 29, further comprising a plurality of cross development environment processes and a router process configured to receive the transaction and route the transaction to one of the cross development environment processes.

31. The apparatus of claim 30, wherein the router process is configured to perform at least one of failover and load balancing in connection with routing the transaction to a cross development environment process.

32. The apparatus of claim 29, wherein the program code is further configured to communicate the transformed transaction to the second software development environment, and to retry communication of the transaction to the second software development environment in response to unavailability of the second software development environment.

33. An apparatus, comprising:

a memory configured to store a mapping data structure for use in a cross development environment that couples together a plurality of software development environments, wherein the mapping data structure includes at least one wildcarded field;

a processor; and

program code configured to transform a transaction generated by a first software development environment among the plurality of software development environments into a format compatible with a second software development environment among the plurality of software development environments using the wildcarded field in the mapping data structure.

34. A computer system comprising:

first and second software development environments coupled to one another by a cross development environment; and

program code resident in the first software development environment and configured to notify the cross development environment of an update made in the first software development environment.

35. The computer system of claim 34, further comprising program code resident in the cross development environment and configured to dynamically modify a mapping between the first software development environment and the second software development environment in response to notification of the cross development environment.

36. The computer system of claim 34, further comprising program code configured to dynamically notify an administrator in response to the update.

37. A program product, comprising:

program code configured to detect an update made in a first software development environment among a plurality of software development environments coupled to one another by a cross development environment, and dynamically modify a mapping between the first software development environment and a second software development environment among the plurality of software development environments in response to the detected update; and

a computer readable signal bearing medium bearing the program code.

38. The program product of claim 37, wherein the signal bearing medium includes at least one of a recordable medium and a transmission medium.